Bioinformatics, 38(21), 2022, 4949–4950
https://doi.org/10.1093/bioinformatics/btac620
Advance Access Publication Date: 12 September 2022
Applications Note

Phylogenetics

ASTRAL-Pro 2: ultrafast species tree reconstruction from multi-copy gene family trees

Chao Zhang [1] **and Siavash Mirarab** [2,*]

[1]Bioinformatics and Systems Biology, UC San Diego, La Jolla, CA 92093, USA and [2]Department of Electrical and Computer Engineering, UC San Diego, La Jolla, CA 92093, USA

*To whom correspondence should be addressed.

Associate Editor: Russell Schwartz

Received on July 21, 2022; revised on September 3, 2022; editorial decision on September 7, 2022; accepted on September 9, 2022

Abstract

Motivation: Species tree inference from multi-copy gene trees has long been a challenge in phylogenomics. The recent method ASTRAL-Pro has made strides by enabling multi-copy gene family trees as input and has been quickly adopted. Yet, its scalability, especially memory usage, needs to improve to accommodate the ever-growing dataset size.

Results: We present ASTRAL-Pro 2, an ultrafast and memory efficient version of ASTRAL-Pro that adopts a placement-based optimization algorithm for significantly better scalability without sacrificing accuracy.

Availability and implementation: The source code and binary files are publicly available at https://github.com/chaoszhang/ASTER; data are available at https://github.com/chaoszhang/A-Pro2_data.

Contact: smirarab@ucsd.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

A main goal of phylogenomics is inferring the species tree using methods that allow for discordance among gene trees. Species tree estimation from single-copy gene trees using summary methods has been a standard part of phylogenomic analyses for close to a decade (Mirarab *et al.*, 2021). For example, the ASTRAL family of methods for finding the tree with the maximum quartet score with respect to the gene trees have been widely used. But since such methods have been mostly constrained to single-copy and putatively orthologous genes, these analyses are often forced to select a subset of data that are available or can be obtained (Smith and Hahn, 2021). Luckily, several recent theoretical and algorithmic advances (Hill *et al.*, 2020; Willson *et al.*, 2022; Yan *et al.*, 2022) have enabled practical species tree estimation from multi-copy gene trees, with the promise of greatly expand the number of loci that can be reliably used for species tree inference (Smith and Hahn, 2021).

Among the methods for multi-copy gene tree inference, ASTRAL-Pro (A-Pro for short) is among the most recent and most accurate (Zhang *et al.*, 2020). The main insight of A-Pro is to extend the notion of the quartet score, the bedrock of ASTRAL, into multi-copy genes in ways that seek to avoid counting quartets driven only by duplication events and avoid double-counting quartets differentiated only by duplication events. Thus, it maximizes an updated measure of quartet similarity called per-locus quartet score. Due to its high accuracy on simulated data and apparent accuracy on real data, it has been quickly adopted.

The scalability of A-Pro needs to improve to keep up with the ever-increasing dataset size. In particular, the memory consumption of A-pro can be prohibitively large (e.g. 80 GB for 500 taxa and 1000 genes). A-Pro adopted the ASTRAL-MP (Yin *et al.*, 2019) code and altered its objective function and search space. While the objective function is computed exactly, the search space formation is heuristic. A-Pro builds its search space by first randomly sampling single-copy trees from multi-copy gene family trees, and then letting ASTRAL complete the single-copy trees and build the search space from the completed trees. The ASTRAL algorithm for completing trees is not optimal for too many missing taxa (Mai and Mirarab, 2022), and in the presence of gene losses, the sampled single-copy trees often have many missing taxa. Thus, to ensure a reasonable accuracy, A-Pro is forced to sample a very large number of single-copy genes, which negatively impacts its running time and memory consumption.

Here, we introduce a completely new implementation of A-Pro called ASTRAL-Pro 2 (A-Pro2 for short) that improves scalability by adopting a new optimization algorithm that we have recently introduced (Zhang and Mirarab, 2022). This algorithm reduces the asymptotic running time growth from quadratic to linear with respect to the number of genes. We show that A-Pro2 maintains the same levels of accuracy while reducing the running time and memory by one to two orders of magnitude.

2 New features and results

The new A-Pro2 algorithm works as follows (see Supplementary Material for details): (i) starting from an empty tree, add species one-by-one with a random order at a place maximizing the

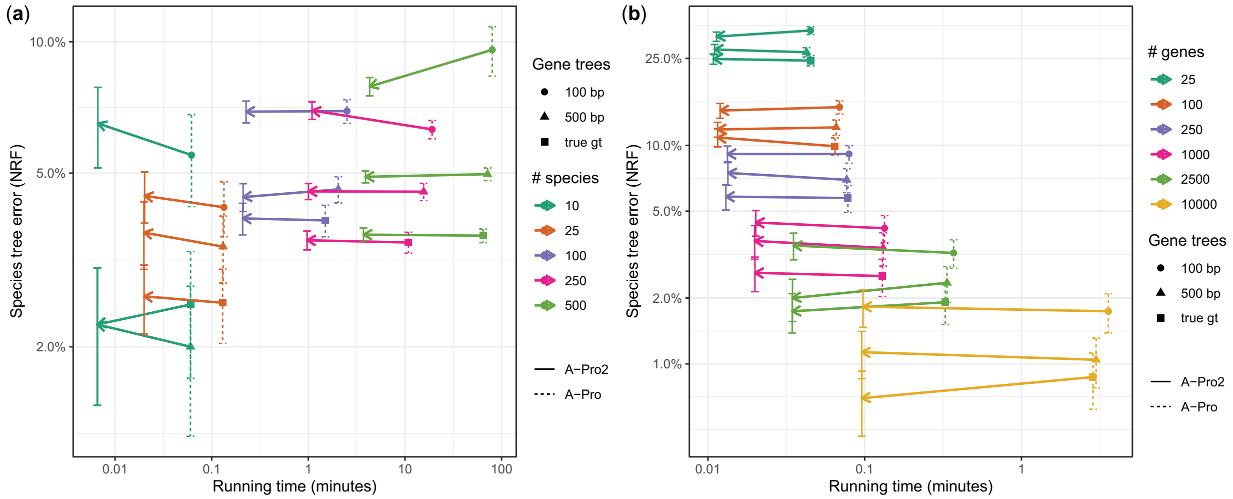

Fig. 1. Species tree error (y-axis) and running time using 28 cores (x-axis) on the S25 dataset as (**a**) the number of in-group species (plus one out-group) or (**b**) the number of gene families changes. Each arrow shows how both metrics change (log-scale) from A-Pro to A-Pro2 (the arrowhead) showing mean and standard error over 50 replicates for each model condition

per-locus quartet score. Use a divide-and-conquer strategy for $n \geq 100$, where n is the number of species. (ii) On the full tree, perform up to $O(\sqrt{n})$ NNI moves to improve its score. (iii) Repeat steps *i* and *ii* for r rounds to build the search space. (iv) Obtain the final tree using a dynamic programming (DP) step with the constraint that each internal node of the output tree is in at least one of the r greedy trees. Thus, the new algorithm circumvents the need to pre-define the search space using single-copy gene tree sampling.

A-Pro2 also includes several new features. (i) *Easy installation*: Original A-Pro requires compiling with Java Native Interface and running with linked binaries. Instead, A-Pro2 is completely implemented in C++ with no dependencies. (ii) *Dynamically adjusted search space*: By default, we set r (which the user can adjust) using a dynamic heuristic: (a) start with four rounds and perform the DP algorithm to get an optimal score; (b) run another four rounds and perform DP using bipartitions from all previous rounds; (c) repeat step (b) until no improvement to the optimal score is obtained or step (b) has been repeated five times. (iii) *Taxon placement*: A-Pro2 natively supports taking a backbone tree as input and efficiently adding taxa onto it. (iv) *Supporting polytomies*: Unlike the original A-Pro that failed when input trees were not binary, A-Pro2 randomly resolves polytomies; it tags resolved nodes as duplication and hence ignores them. (v) *Better parallel efficiency*: A-Pro2 can efficiently parallelize with at least 32 cores on large data. In fact, by efficiently utilizing shared cache, A-Pro2 often experiences superlinear speedup (Supplementary Fig. S1).

We reanalyzed the yeast and plant datasets studied in the original A-Pro paper and observed no topological changes in either case. For the larger plant dataset with 83 species and 9237 gene trees, running time decreased from 33 to 1 min. We then compared A-Pro2 to A-Pro on the S25 dataset from Zhang *et al.* (2020) and S100 dataset from Molloy and Warnow (2020) (see Supplementary Material for a description of data). We found no significant difference in species tree error ($P = 0.944$ for S25 and $P = 0.674$ for S100; Fig. 1 and Supplementary Fig. S2).

A-Pro2 has much lower running time compared to original A-Pro, especially for the larger input data (Fig. 1). For example, A-Pro2 achieves 18× speedup on average on 501-taxon datasets with 1000 gene families, going from 1.2 h to 4 min (Fig. 1a); also, it produces a 32× speedup with $25 + 1$ species and 10 000 gene families (Fig. 1b). The error reduces from 9.6% to 7.9% for $500 + 1$ species with 100 bp genes but slightly increases from 5.5% to 6.5% for $10 + 1$ species.

Memory was the main limitation of A-Pro. Adopting the new optimization algorithm and also the switch to C++ (instead of garbage collection used by Java) significantly reduced memory requirement of A-Pro. For example, on S25 dataset with $500 + 1$ species and 1000 gene families, A-Pro2 allocates only 2.3 GB of memory, which is $<$ 3% of what original A-Pro requires (84 GB).

Funding

This work was supported by the National Science Foundation, Award Number IIS-1845967 to S.M.

Conflict of Interest: none declared.

References

Hill,M. *et al.* (2020) Species tree estimation under joint modeling of coalescence and duplication: sample complexity of quartet methods. *arXiv*, **2007**, 6697. https://doi.org/10.48550/arXiv.2007.06697.

Mai,U. and Mirarab,S. (2022) Completing gene trees without species trees in sub-quadratic time. *Bioinformatics*, **38**, 1532–1541.

Mirarab,S. *et al.* (2021) Multispecies coalescent: theory and applications in phylogenetics. *Annu. Rev. Ecol. Evol. Syst.*, **52**, 247–268.

Molloy,E.K. and Warnow,T. (2020) FastMulRFS: fast and accurate species tree estimation under generic gene duplication and loss models. *Bioinformatics*, **36**, i57–i65.

Smith,M.L. and Hahn,M.W. (2021) New approaches for inferring phylogenies in the presence of paralogs. *Trends Genet.*, **37**, 174–187.

Willson,J. *et al.* (2022) DISCO: species tree inference using multicopy gene family tree decomposition. *Syst. Biol.*, **71**, 610–629.

Yan,Z. *et al.* (2022) Species tree inference methods intended to deal with incomplete lineage sorting are robust to the presence of paralogs. *Syst. Biol.*, **71**, 367–381.

Yin,J. *et al.* (2019) ASTRAL-MP: scaling ASTRAL to very large datasets using randomization and parallelization. *Bioinformatics*, **35**, 3961–3969.

Zhang,C. and Mirarab,S. (2022) Weighting by gene tree uncertainty improves accuracy of quartet-based species trees. *bioRxiv*, page 2022.02.19.481132. https://doi.org/10.1101/2022.02.19.481132.

Zhang,C. *et al.* (2020) ASTRAL-Pro: quartet-based species-tree inference despite paralogy. *Mol. Biol. Evol.*, **37**, 3292–3307.